WEST

☑ Generate Collection Print

L3: Entry 5 of 8 File: PGPB Jul 11, 2002

DOCUMENT-IDENTIFIER: US 20020091860 A1
TITLE: Relocating context information in header compression

Summary of Invention Paragraph (6):
[0005] For this purpose, IETF (Internet Engineering Task Force) has lately been working on the standardisation of a header field compression method known as ROHC (Robust Header Compression). One idea behind the development of ROHC is that there is a great deal of redundance between the several IP header fields used in data packet transfer, not only inside the data packet, but also between them. In other words, a large amount of the information in the header fields does not change at all during the transfer of the data packets and is thus easy to reconstruct at a receiver even though it is not transmitted. Only a small part of the header fields are such that the information they comprise requires attention during compression. Further, ROHC comprises several compression levels, whereby the efficiency of the compression increases when moving on a higher level. ROHC always tries to use the most efficient compression possible, in such a manner, however, that before moving on to the next level, a sufficient reliability of operation of the level is always ensured. ROHC also has the typical characteristic that it leaves several matters essential for the use of a compression method to be handled by the lower link layer.

Detail Description Paragraph (2):
[0022] In the following, the invention is illustrated by the way of an example in conjunction with the header field compression method ROHC, which is particularly suitable for real-time data transfer over a radio interface. The invention is not limited to only ROHC, but it can be applied to any other header compression method, too. Furthermore, the invention is applicable to, but not limited to IP/UDP/RTP header compression. In the following, the implementation of ROHC is described for the parts essential for the invention. For a more detailed description of the compression method in question, reference is made to a yet unfinished Internet draft "Robust Header Compression (ROHC)", version 04, Oct. 11, 2000.

Detail Description Paragraph (35):
[0055] When the radio bearer for packet-switched user data is established (RB establishment) or reconfigured between the mobile terminal and the radio network, both peers negotiate the parameters of the radio bearer using signalling according to a radio resource control protocol RRC. The radio resource control protocol RRC is responsible e.g. for establishing, configuring, maintaining and terminating radio connections between the mobile terminal and the radio network UTRAN and for transmitting control information transmitted from the core network CN and the radio network RAN to the mobile terminals UE. One of the parameters defining the radio bearer is the header compression method used by the terminal. Compressing the headers of data packets to be transmitted and decompressing received data packet headers is in the UMTS system performed on the packet data convergence protocol PDCP layer belonging to the packet data protocol. The tasks of the PDCP layer include functions related to improving channel efficiency, which are typically based on different optimisation methods, such as the utilisation of data packet header compression algorithms. Because currently the network-level protocols designed for UMTS are IP protocols, the compression algorithms used are those standardised by IETF (Internet Engineering Task Force). Thus, the ROHC compression method is especially well-suited for the UMTS system.

CLAIMS:

10. A method in accordance with claim 1 wherein: said method is used in accordance with Robust Header Compression (ROHC) implemented in a UMTS system.

21. A packet network in accordance with claim 12 wherein: said packet network is a UMTS system, wherein Robust Header Compression (ROHC) is implemented.

L3: Entry 7 of 8 File: PGPB Jan 3, 2002

DOCUMENT-IDENTIFIER: US 20020001315 A1
TITLE: Method and apparatus for compressing IP/UDP/RTP headers in a lossy environment

Abstract Paragraph (1):
A system and method for compressing data packet headers in a lossy environment comprises circuitry that initially transmits a data packet with a full header and subsequently transmits data packets with compressed headers. The compressed headers include information that details the differences between the subsequent packets and the original full packet. Accordingly, each subsequent packet has a header whose differences are based upon the original full header of the first packet. Accordingly, a base station transceiver system includes circuitry for transmitting and receiving data packets utilizing the described header compression algorithm. Similarly, a mobile terminal is formed to receive and transmit data packet headers compressed in this manner.

Summary of Invention Paragraph (5):
[0005] Real time applications over lossy links having bandwidth constraints present design issues with competing interests. For example, that the link is a lossy link suggests that robust signal transmissions with significant overhead may improve accurate signal delivery. The types of mechanisms that improve signal delivery, however, require additional scarce communication resources. On the other hand, if transmission mechanisms are implemented that minimize the amount of scarce resources required to deliver the signal, then the information delivery is more subject to interference in a lossy environment. As a result of these design issues, there has been a great effort to develop methods of transmitting communication signals in a lossy environment that is both robust and maximally efficient.

Summary of Invention Paragraph (6):
[0006] As one example, designers have developed various header compression schemes for packets transmitted over a wireless interface. One common header compression technique includes transmitting a first full header and then, for a second header, transmitting a header that is compressed and that includes the differences between its full header information and that of the previously transmitted full header. Thereafter, a third header includes information that details the differences between the third and the second headers. The above described method for header compression is advantageous in that, for example, a 40 byte header may be reduced to 4 bytes in size in an optimal situation. The disadvantage to this scheme, however, is that an error in the transmission of the second header is propagated to the third, and then the fourth and all subsequent headers. What is needed, therefore, is a header compression scheme that achieves the efficiencies of known header compression schemes but that provides a more reliable and robust operation.

Detail Description Paragraph (4):
[0023] As was described previously, formats such as those seen in IP version 6, more specifically, data packet 132, support robust system operation. In a lossy environment, and in a wireless environment where resources are a premium, such overhead signaling consumes a very significant portion of the air resources thereby limiting the number of users at any given time. Accordingly, there is a significant need to reduce the signal size to increase capacity.

Detail Description Paragraph (6):
[0025] In theory, the system of FIG. 2 works ideally because the headers may be compressed to only define the differences between a present header and a previous header thereby avoiding the transmission of unnecessary signaling. In theory, if an error occurs thereby impacting one or more headers, once an accurate header is received, the erroneous headers may be repaired and reconstructed. To do so, however, a

robust coding algorithm such as a turbo coding algorithm is preferably used to facilitate data packet repair.

Detail Description Paragraph (7):
[0026] In a lossy environment, however, significant system resource becomes dedicated to repairing and to enabling the repair of headers. Additionally, headers are not as easily repaired as theory might suggest. Accordingly, a better header compression scheme is necessary. The problem becomes especially acute because, for example, if a transmission error occurs for packet P.sub.1, then the packets P.sub.2 and P.sub.3 also are not properly reconstructed unless a robust and resource hungry coding scheme is used. For example, packet P.sub.2 includes a header that defines the differences between it and the header of the previous packet P.sub.1. If a robust coding scheme is not used, however, subsequent packets are affected if a packet is received in error. For example, if packet P.sub.1 is received with errors packet P.sub.2 cannot be properly reconstructed. Similarly, because packet P.sub.2 cannot be properly reconstructed, packet P.sub.3 cannot be properly reconstructed. As may be seen, therefore, the transmission error of one packet has a domino affect on the reconstruction of packets for subsequent packets. What is needed, therefore, is a system that achieves the same efficiencies of the header compression scheme of FIG. 2 but that avoids the domino affect that results from the erroneous receipt of a given data packet and that avoids a need for using robust coding on al data packets.

Detail Description Paragraph (9):
[0028] FIG. 4 is a signal diagram illustrating a format of a data packet having a full header and the data packet having a compressed header. A data packet shown generally at 404 includes a first portion 408 that is for defining whether the packet is a compressed packet or uncompressed packet. For example, a 1 might be used to indicate that the following packet is in a compressed format while a 0 might represent that it was transmitted in an uncompressed format. A second portion 412 then carries the full value of the header plus necessary coding. Here, a robust type of coding, such as turbo coding, is used to ensure the accurate delivery of the first packet.

WEST

End of Result Set

✓ Generate Collection Print

L3: Entry 8 of 8 File: USPT Nov 5, 2002

DOCUMENT-IDENTIFIER: US 6477669 B1
TITLE: Method and apparatus for adaptive control of forward error correction codes

Brief Summary Text (9):
In contrast, the bit error ratios associated with wireless communication are much higher (on the order of 10.sup.-3 to 10.sup.-8) and tend to fluctuate in accordance with atmospheric conditions. In addition, the errors associated with wireless communication tend to occur in longer bursts. Thus, a robust error correction scheme must be employed in a wireless network in which ATM based technology is to be implemented.

Brief Summary Text (15):
As explained in the Provisional Application Ser. No. 60-052,359, which is incorporated herein by reference, the frame relay system uses a robust, flexible frame format between two communicating terminals which allows the transport of several variable sized Spackets (segmented packets) in a frame and also allows a single Spacket to be carried over several frames, whichever the case might be. Also, the frame format allows fast synchronization and the exchange of coding information. Each frame contains Reed-Solomon (RS) check bytes that are used for error correction and to enhance the quality of the satellite/wireless link. The number of RS check bytes in a frame can be changed on the fly, without any loss of data, to compensate for varying link conditions. The decision to change the RS check bytes in a frame is based on the constant monitoring of the link quality. Several frames are also interleaved before transmission over the satellite/wireless link, to help spread the effect of burst errors over several frames, all of which can then be corrected by the FEC in the frames.

Detailed Description Text (23):
If the value of FRAMENUM is equal to 1, then the least significant bit of the coding field 268 is set to 1 to represent the fact that an ATM cell header compression algorithm has been activated. If the value of FRAMENUM is equal to 2 or 3, then the coding field 269 may be set to zero unless Viterbi coding is utilized, as noted below.

Detailed Description Text (80):
The illustrated system uses a robust, flexible frame format between the 2 communicating terminals which allows the transport of several variable sized Spackets (segmented packets) in a frame and also to carry a single Spacket over several frames, whichever the case might be. Also, the frame format allows fast synchronization and the exchange of coding information. Each frame contains Reed-Solomon check bytes that are used for error correction and to enhance the quality of the satellite/wireless link. The number of RS check bytes in a frame can be changed on the fly, without any loss of data, to compensate for varying link conditions. The decision to change the RS check bytes in a frame is based on the constant monitoring of the link quality. Several frames are also interleaved before transmission over the satellite/wireless link, to help spread the effect of burst errors over several frames, all of which can then be corrected by the FEC in the frames. Also, Virtual Channels (VCs) can be configured to be enabled for data compression, which means that the Spackets belonging to the VC are to be passed through a data compressor/decompressor combination to save bandwidth. VCs can also be configured to be either high or low priority VCs and the scheduler then, uses this information to fairly transmit the various Spackets over the satellite/wireless link.

WEST

Print Selection

Help | Clear | Cancel | Print | Print First Page

Select?	Document ID	Section(s)	Page(s)	# Pages to print	Database
✓	20020091860	all	all	17	USPT,PGPB,JPAB,EPAB,DWPI
✓	20020001315	all	all	15	USPT,PGPB,JPAB,EPAB,DWPI
✓	6477669	all	all	18	USPT,PGPB,JPAB,EPAB,DWPI

Building	Room	Printer
cpk2	4c32	gbfrptr

Main Menu | Logout

WEST

✓ Generate Collection Print

L6: Entry 1 of 13 File: PGPB Jan 9, 2003

DOCUMENT-IDENTIFIER: US 20030007490 A1
TITLE: Packet data service in radio communication system

Summary of Invention Paragraph (37):
[0035] The PDCP PDU is transmitted to the UE by passing through the RLC, MAC, and L1. The transmitted PDCP PDU is delivered to the PDCP of the UE by passing through the L1, MAC, and RLC in the UE. Then, a header decompression is made using reverse algorithm to that of the header compression. Then, the extracted PDCP SDU is transferred to the upper layer (PPP, IP). IP packets from UE side can be transmitted to the UTRAN side by similar manner.

Summary of Invention Paragraph (64):
[0062] In step 10, the UE RRC notifies the RRC in the target RNC of the DL RSN. The target RRC compares the DL RSN transmitted from the UE to the DL SSN transmitted from the source RRC to decide the first DL PDCP SN of the PDCP SDU which will be transmitted to the UE at next time.

Summary of Invention Paragraph (72):
[0070] Moreover, a problem of modular comparison which is generated when the target RRC compares the SN can not be solved yet.

Detail Description Paragraph (6):
[0117] The PDCP SDU buffer is preferably used instead of the PDCP PDU buffer. If the LSR is generated, a header compression algorithm is newly assigned, and therefore, the algorithms used in a source PDCP and used in a target PDCP are differentiated from each other. Therefore, the PDCP PDUs which were compressed using the algorithm before the LSR is performed cannot be decompressed after the LSR process is ended.

Detail Description Paragraph (7):
[0118] That is, the PDCP SDUs are stored in the buffer, and then are forwarded to the target PDCP in the state that the SDUs are not compressed when the LSR is generated. And the target PDCP compresses and transmits the forwarded SDUs using the newly assigned header compression algorithm.

Detail Description Paragraph (8):
[0119] As shown in FIG. 10, in case that the PDCP supports the LSR, when the PDCP receives the SDUs, the PDCP stores the SDUs in the buffer as respective SDU unit form, after that, the PDCP performs the header compression according to the provided header compression algorithm to generate the PDCP PDUs (=RLC SDUs), and descends the PDCP PDUs to an RLC AM entity.

Detail Description Paragraph (34):
[0145] The third case is needed in following case. The target RRC decides the first DL PDCP SN of the PDCP PDU which will be transmitted first at the next time by comparing the DL SSN received from the source RRC and the DL RSN received from the UE RRC during the LSR process (step 11 in FIG. 9a).

Detail Description Paragraph (41):
[0152] The above problem is generated because the target RRC does not know a valid range of the DL RSN. In addition, modular comparison problem is generated in the above case.

Detail Description Paragraph (70):
[0181] In addition, the header compression algorithms used before and after the LSR may be different from each other, and the feedback PDU, which is not confirmed, includes out-of-date information. Therefore, the feedback information is not transmitted to the

target PDCP and discarded at the source PDCP.

WEST

✓ Generate Collection Print

L6: Entry 2 of 13 File: PGPB Oct 3, 2002

DOCUMENT-IDENTIFIER: US 20020142757 A1
TITLE: Method and apparatus for broadcast signaling in a wireless communication system

Detail Description Paragraph (40):
[0079] In one embodiment, the SPI approach applies several criteria. Firstly, a single CS uses the same protocol options for consecutive streaming sessions, else the CS modifies the SPI when the protocol options change. Secondly, the PDSN does not change the header compression algorithm or the parameters between streaming sessions with the same SPI.

Detail Description Paragraph (62):
[0101] The use of a BLOB of information is illustrated in FIG. 20, wherein a broadcast session is assigned a set of parameters. Each parameter may be one of multiple options. The transmission of the parameters provides a level of flexibility in comparison to the use of fixed sets of parameters associated with a SO number. The CS may select any of the available options, and transmits the information to the MS. As illustrated, the FIELD 2 of the BLOB may be specified as any of the options: OPTION 1 to OPTION K, wherein each field of the BLOB may have a different number of available options.

WEST

☑ Generate Collection Print

L6: Entry 5 of 13 File: PGPB Sep 12, 2002

DOCUMENT-IDENTIFIER: US 20020126675 A1
TITLE: Packet transmission method and system, and packet transmitting apparatus, packet receiving apparatus, and packet transmitting/receiving apparatus

Detail Description Paragraph (34):
[0088] Thus, according to this embodiment, the scheduling process is performed at the data link layer, while the conventional scheduling process is performed at the IP layer that is an upper layer compared to the data link layer. Also, in the data link control process, by dividing each transmission packet into a plurality of shorter data units, and by scheduling the data units, when the real-time type packet comes to be transmitted during transmission of the data type packet, since the length of one unit of transmitted data (that is the above predetermined data unit) is shorter than one of the conventional transmitted packet, the waiting time for a real-time type packet to be transmitted can be reduced.

Detail Description Paragraph (51):
[0105] The header compression parts 1301 may employ arbitrary header compressing algorithm as the header compression process, such as RFC1144, RFC2507, and RFC2508 of IETF (Internet Engineering Task force), or RFC3095.

WEST

✓ Generate Collection Print

L6: Entry 8 of 13 File: PGPB May 30, 2002

DOCUMENT-IDENTIFIER: US 20020064190 A1
TITLE: Method for compressing packet headers within a trunking protocol for aggregating multiple information channels across a network

Detail Description Paragraph (17):
[0043] Comparing packet 30 to packet 24 shown in FIG. 3A, the percentage overhead is reduced from 3H/P to (2C+H)/P. For a packet payload of 24 bytes and a header of 20 bytes, this results in a reduction in overhead from 250% in case of packet 24 shown in FIG. 3A to 100% in case of packet 30, smaller than the 166% overhead without the trunking protocol. This dramatic improvement enables the trunking protocol to take full advantage of the aggregation of multiple information channels, since the effective bandwidth is now able to approximately reach the maximum bandwidth that can be achieved by the aggregated channels. If three information channels are aggregated with each information channel providing a maximum bandwidth equal to B, for example, the header compression technologies now enable the trunking protocol to achieve an effective bandwidth approximately equal to 3B.

Detail Description Paragraph (22):
[0048] Comparing packet 32 to packet 25 shown in FIG. 3B, the percentage overhead is reduced from 6H/T to (2C+4H)/P. For a packet payload of 1000 bytes, H equal to 20 bytes, and C equal to two bytes, this results in a reduction in overhead from 12% in case of packet 25 shown in FIG. 3A to 8.4% in case of packet 32, as opposed to 4% without the trunking protocol. This dramatic improvement enables the trunking protocol to take full advantage of the aggregation of multiple information channels, since the effective bandwidth is now able to more closely approximate the maximum bandwidth that can be achieved by the aggregated channels.

CLAIMS:

8. The method of claim 1, wherein applying header compression to the inner packet headers comprise applying the RFC 2508 header compression algorithms to the inner packet headers.

21. The method of claim 14, wherein applying header compression to the inner packet headers comprise applying the RFC 2508 header compression algorithms to the inner packet headers.

WEST

☑ Generate Collection Print

L6: Entry 9 of 13 File: PGPB Jan 3, 2002

DOCUMENT-IDENTIFIER: US 20020001315 A1
TITLE: Method and apparatus for compressing IP/UDP/RTP headers in a lossy environment

Abstract Paragraph (1):
A system and method for compressing data packet headers in a lossy environment comprises circuitry that initially transmits a data packet with a full header and subsequently transmits data packets with compressed headers. The compressed headers include information that details the differences between the subsequent packets and the original full packet. Accordingly, each subsequent packet has a header whose differences are based upon the original full header of the first packet. Accordingly, a base station transceiver system includes circuitry for transmitting and receiving data packets utilizing the described header compression algorithm. Similarly, a mobile terminal is formed to receive and transmit data packet headers compressed in this manner.

Detail Description Paragraph (21):
[0040] FIG. 8 is a flow chart illustrating the method for receiving and interpreting compressed headers according to one embodiment of the present invention. Initially, the inventive process includes receiving a data packet (step 804). Thereafter, the system determines the type of packet that was transmitted in a compressed or uncompressed (Full) format (step 808). If a packet with a full header was transmitted, then the header is stored (step 812). If a compressed header was transmitted, the compressed header is compared to the full header (step 816). Thereafter, the compressed header, more particularly, the changes from the full header, as defined in the compressed header, are used to reconstruct the full header for the received data packet (step 820). Once the full data packet has been reconstructed in step 820 or after it has been stored in step 812, the signal is processed in a routine manner (step 824). The above method steps may be performed either by a mobile terminal or by a base station transceiver system.

Detail Description Paragraph (22):
[0041] FIG. 9 is a flow chart illustrating a method for processing received data packets according to an embodiment of the present invention. The method is performed in a receiver, which receiver can comprise either a base station transceiver system or a mobile terminal. Initially, the receiver receives a full header and stores the same (step 904). Thereafter, it receives a second header (step 908) and compares the changes in the second header to the full header to reconstruct the header of the second packet (step 912). The reconstructed packet is then processed (step 914).

Detail Description Paragraph (23):
[0042] Thereafter, the receiver receives and stores a third header, which header is compressed and specifies changes relative to the full header of the first data packet (step 916) and compares the third compressed header differences to the full header of the first packet (that was previously stored) to reconstruct the header of the third packet (step 920). Thereafter, the reconstructed packet is processed using conventional processing techniques (step 922). If the receiver is a mobile terminal, then the received packets are reconstructed and converted to audio by an audio processor (step 924) and are produced to a speaker for playback to the user (step 928).

CLAIMS:

1. A method for transmitting data packets in a lossy environment, comprising: transmitting a first data packet with a full header; transmitting a second data packet with a compressed header, which compressed header includes differences based upon the full header of the first data packet; and transmitting a third data packet with a compressed header, the compressed header of the third data packet including differences

in the header of the third data packet as compared to the first data packet.

WEST

☑ Generate Collection Print

L6: Entry 10 of 13 File: USPT Mar 18, 2003

DOCUMENT-IDENTIFIER: US 6535925 B1
TITLE: Packet header compression using division remainders

Brief Summary Text (5):
Conventional header compression algorithms are designed basically for narrow band wired channels with a rather small complexity at the receiving decompression side. Also, the complexity at the sending compressing side is kept low to allow efficient implementations in routers where as much computing capacity as possible is needed for the routing. Further, the wired channels for which existing header compression algorithms are designed typically have very small probabilities for bit errors (e.g., a bit error rate of 10.sup.-6). Wireless channels (generally characterized by lossy, narrow bandwidth links) typically have a much higher probability for error, so header compression for use in wireless channels should be designed with a much larger bit error probability in mind (e.g., bit error rates up to 10.sup.-3).

Detailed Description Text (23):
A verifier 79 receives as input TS_guess and the received version of the checksum from extractor 72. The verifier 79 is operable to generate a checksum from the received TS_guess value and (optionally) other information received in the compressed header 22' (see broken line), and compare this generated checksum to the received checksum. If the checksums match, then the verifier output signal 704 activates a connection unit 701 which then connects the TS_guess value to selector 700.

Detailed Description Text (28):
FIG. 8 illustrates exemplary time stamp decompression operations which can be performed by the time stamp decompressor embodiments of FIGS. 6-7A. It is first determined at 80 whether or not the time stamp field includes the resume code. If not, then the time stamp field is decompressed using conventional decompression techniques at 81, and the next packet is then awaited at 89. If the resume code is detected at 80, then the time stamp estimate (TS_estimate) is calculated at 82 (with scaling as desired), and the most significant bits are extracted therefrom at 83. At 84, the least significant bits received in the compressed header are appended to the most significant bits extracted from the scaled estimate, and the result is (re-scaled as necessary) is the time stamp guess (TS_guess). Thereafter at 85, the time stamp guess is used to generate a checksum, and the generated checksum is compared at 86 to the checksum received in the time stamp field. If the generated checksum matches the received checksum, then the time stamp guess is accepted at 87, and the next packet is then awaited at 89. If the generated and received checksums do not match at 86, it is then determined at 88 whether or not to give up estimating the time stamp, for example, based on a predetermined elapsed time value, or a predetermined number of guesses. If it is decided not to give up at 88, then another scaled time stamp estimate is calculated at 82, and the operations at 83-86 are repeated. In making another time stamp estimate, the estimator 75 can, for example, change one or more of the least significant bits of the MSBs that will be extracted from the estimate. In one example, if changing a particular bit (or bits) results in successful re-estimation of the time stamp of a given packet, then this same change can be tried first when re-estimating the time stamp of a subsequent packet. If it is decided to give up at 88, then the next packet is awaited at 89.

Detailed Description Text (37):
Positive values of M can, for example, be used effectively to accommodate sequence number fields (for example RTP sequence number fields) of packets that arrive out of order. Positive values of M are also advantageous, for example, to accommodate negative delta time stamp field values for packets including so-called B-pictures (bidirectionally predicted pictures in an MPEG application). In MPEG, a B-picture is conventionally sent after its temporarily surrounding anchor pictures, thus leading to

both forward and backward jumps in the time stamp field value as compared to the transmission order. Thus, the time stamp delta is sometimes negative and sometimes jumps forward, which is a well known phenomenon to workers in the art. As mentioned above, positive values of M can accommodate negative deltas. Selecting M such that the range includes 0 permits accommodation of pictures that are partitioned into many packets with the same time stamp field.

Detailed Description Text (39):
A comparator 123 compares the received remainder at 118 to the remainders of each of the field value candidates. Because there are X adjacent field value candidates in the range, and because the received remainder at 118 represents the remainder of a divide by X operation, the remainder of one of the X field value candidates will match the received remainder 118, and the corresponding field value candidate is output at 122 from buffer 128. As shown in FIG. 12, the candidate value output at 122 can be scaled up as necessary to accommodate any downscaling that may have been done in the header compressor of FIG. 10. Such upscaling can include adding the remainder that results from the downscaling division operation in the header compressor, which remainder is generally a constant that need only be transmitted once, for example, by transmitting the full field value at the start of the packet stream, thereby implicitly informing the header decompressor of the downscaling remainder. In the FIG. 12 embodiment, the candidate value at 121 (scaled or not) can be output as the reconstructed field value 111 of FIG. 11. This reconstructed field value is also stored in buffer 115 for use as the latest reconstructed field in the next header field reconstruction operation.

Detailed Description Text (42):
where the first number in each set of parentheses is the full 16 bit field value, and the second number is the field value modulo 12. If the latest reconstructed field value is 65531, and the received remainder is 1, and assuming for this example that M=-1, then two of the twelve candidate field values, namely 65533 and 1, will have remainders that match the received remainder in the comparator 123 of FIG. 12.

Detailed Description Text (43):
One exemplary solution to this problem is to use the received checksum to verify which of the two possible candidates is correct. As shown in the embodiment of FIG. 13, a verifier 136 can be coupled to the (scaled or not) buffer output 121 to receive the two matching candidate field values output from buffer 128. The verifier 136 can compute a checksum for each candidate value, compare it to the received checksum, and select the candidate whose checksum matches the received checksum. This candidate can then be provided as the reconstructed field value at 111.

Detailed Description Text (50):
FIG. 14 illustrates exemplary operations which can be performed by the header decompressor embodiments illustrated in FIGS. 11-13. At 141, the compressed header field is received as a modulo X remainder value. At 142, the range information and the latest reconstructed field value are used to generate the candidate field values. At 148, the modulo X remainders of the candidate values are determined. At 143, the received remainder value is compared to the remainder values of the respective candidate field values to determine the matching candidate(s). It is then determined at 144 whether or not there are plural matching candidates. If so, the candidate values are subjected at 145 to the checksum verification process to determine the correct candidate value, which is then loaded as the reconstructed field value at 149. If there is only one matching candidate at 144, then that candidate value can be optionally corroborated by checksum verification at 147 or, as indicated by dotted lines, the candidate can at 149 be directly loaded as the reconstructed field value.

CLAIMS:

12. The method of claim 4, wherein said selecting step includes comparing the received compressed header field to compressed header fields respectively associated with the reconstructed header field candidates, and selecting as the reconstructed header field one reconstructed header field candidate whose associated compressed header field matches the received compressed header field.

WEST

[✓] Generate Collection Print

L3: Entry 1 of 3 File: USPT Sep 3, 2002

DOCUMENT-IDENTIFIER: US 6446068 B1
TITLE: System and method of finding near neighbors in large metric space databases

Detailed Description Text (38):
Note that it is possible, given the link structure, that an item in the current search set will have a link to another item which was or is already in the current search set. This could result in computing the same query-to-item distance multiple times. This may be easily avoided by maintaining, during each search, a list of all items which have been compared to the query, along with the corresponding distances. Well-known hash table techniques can allow fast access to items in such a list. If a linked item is already in this list, its distance to the query may simply be read from the list rather than re-computed.

Current US Original Classification (1):
707/6

Current US Cross Reference Classification (5):
707/104.1

Current US Cross Reference Classification (6):
707/4

Current US Cross Reference Classification (7):
707/5